

Providing QoS Assurance and Mobility Support on Linux: RSVP, Mobile IP and CBQ

Presented by
Dr KC Chua,

Centre for Wireless Communications,
National University of Singapore

CC Foo, KJ Loh

Department of Electrical Engineering,
National University of Singapore

Tutorial Outline

- Background and Motivation
 - What is QoS?
 - What is Mobility?
- Resource Reservation
 - Introduction on RSVP
- Mobility Support
 - Introduction on Mobile IP

Tutorial Outline

- Implementation of RSVP on Linux
- Implementation of MIP on Linux
- Providing QoS in Mobile Environment
- Implementation of CBQ on Linux
- Performance of CBQ

Existing Internet Model

• Strengths

- Support internetworking among diverse collections of hardware and networks.
- Distributed administration of sub-components.
- Applications independent of underlying networks.
- Best effort datagram model is simple and robust as intelligence is built in end-hosts for packet loss detection and recovery, as well as delay adaptation.

Existing Internet Model

- Weaknesses
 - Unable to provide different grades of service.
 - Unable to support guaranteed level of performance.
 - Routers based on old packet scheduling technology.
 - Hierarchical addressing scheme not conducive to mobility.

Why is QoS Important?

- The golden rule of disk space applies to bandwidth too: no matter how much you have, it's 90% utilised. Hence the need for arbitration of competing resource demands.
- Real-time applications require some service guarantees. There is a limit as to how much the applications can adapt or recover from packet loss.

Why is QoS Important?

- Users should not be treated equally. Service accorded must correspond to their needs and willingness to pay.
- Future applications need better service model than the simple best-effort model.

Why is Mobility Important?

- People want the ability to access computing and networking resources whenever and wherever they are.
- Evidenced by the popularity of laptops and PDAs.
- Evidenced by industry interest in providing wireless data services eg GPRS (GSM), CDPD (AMPS), Metricom.

Why is Mobility Important?

- Improve the quality of people's life, and also
- New business opportunities,
- such as :
- Location-based services.
- On-the-move collaborative and communication tools.
- Access corporate databases on the field.

Integrated Services Internet Model

- The next generation Internet that supports classical data and real-time multimedia traffic.
- Provides different grades of services, e.g. Guaranteed Service, Controlled-Load Service.
- Resource reservation achieved by using RSVP.

Elements in Integrated Services Architecture

- Service Models:- Guaranteed Service, Controlled Load Service, Best Effort Service
- Reservation Setup Protocol (RSVP)
- Admission Control
 - decides whether to accept or reject a reservation request depending on resources available and policy.

1-5 March 1999

Singapore Linux Conference

11

Elements in Integrated Services Architecture

- Packet Scheduler (WFO, CBO)
 - Decides the sequence of packets to be forwarded on a link. Controls the resources consumed by flows.
- Packet Classifier
 - Maps incoming packets into classes. Different classes get different treatment from packet scheduler. Presently, packets are classified by IP header. May classify by content types.

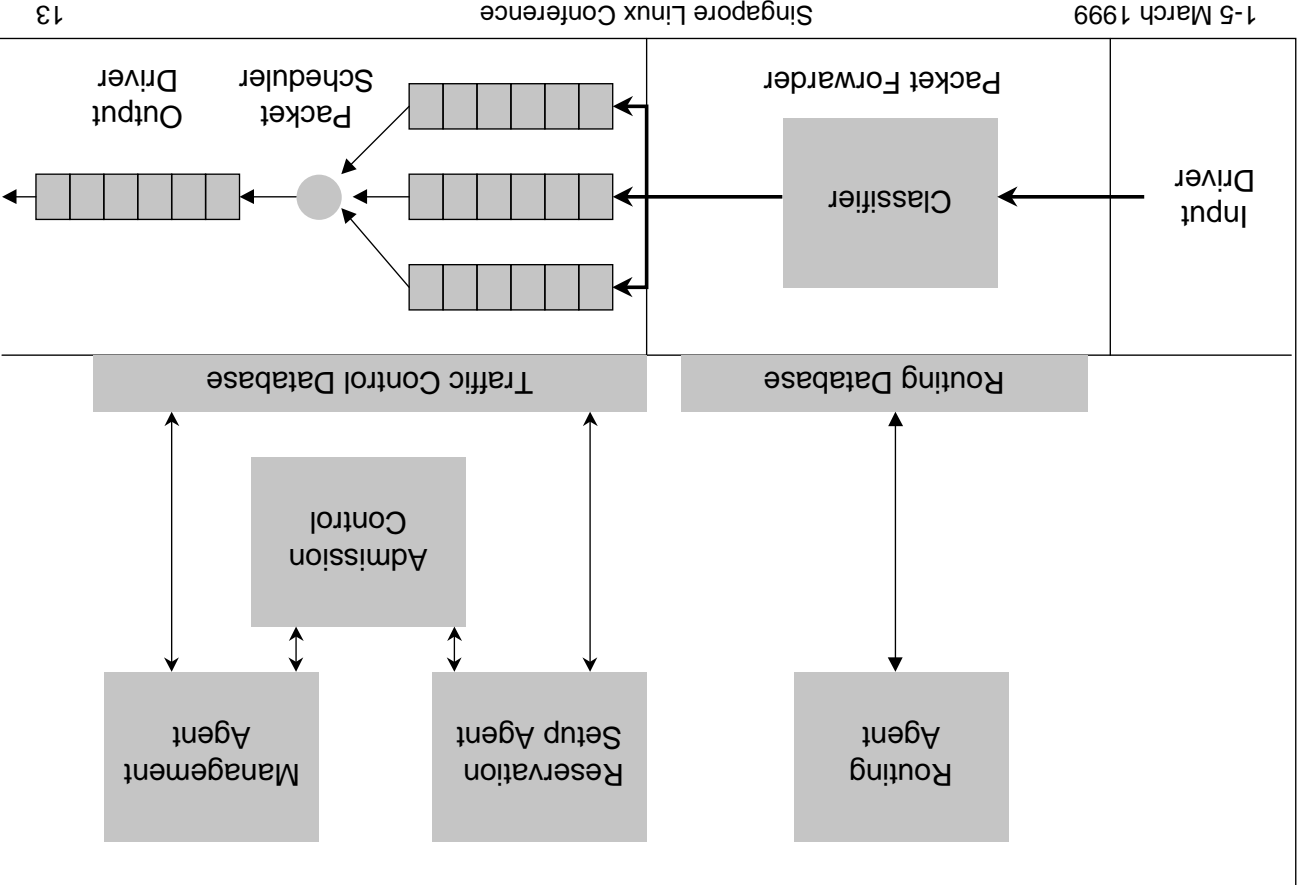
1-5 March 1999

Singapore Linux Conference

12

Types of Service

- **Guaranteed Service**
 - Provides assured level of bandwidth, a firm end-to-end delay bound, and no queueing loss for conforming packets of data flow. Intended for applications that need hard delay bound.
- **Controlled Load Service**
 - Provides no firm guarantees. Makes commitment to offer the flow a service equivalent to that seen by a best-effort flow on a lightly loaded network.



RSVP

- A messaging protocol that enables senders, and receivers to specify their resource requirements and convey them to the intermediate routers.
- It facilitates the setting up and maintenance of reservation along the path(s) that data flow has taken.
- It allows routers to be aware of the reservation requests from end hosts.

Salient Features of RSVP

- Receiver driven reservation.
- Supports both unicast and multicast sessions.
- Supports heterogeneous reservation.
- Uses soft state (time-outs and refreshes).
- Supports various reservation styles (WF, SE, FF).
- Makes unidirection reservation.
- One pass with advertisement.

Typical Scenario

- A sender application conveys its traffic characteristics downstream using PATH message.
- Intermediate routers install PATH states and optionally update the PATH messages with their link states.
- Upon receiving PATH, receivers decide the amount of resources to reserve. This info is sent upstream using RESV.

1-5 March 1999

Singapore Linux Conference

17

Typical Scenario

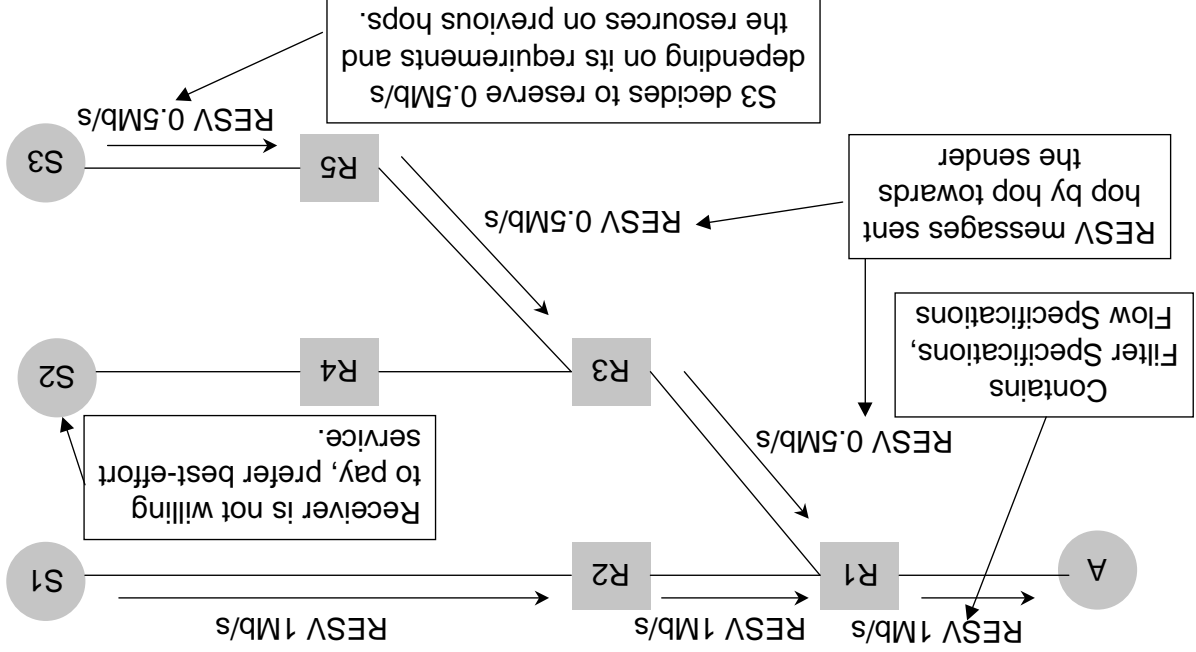
- At each hop, RESV states are installed and resources made available for the flow.
- If there are not enough resources at a hop, the receivers will be informed by the router.
- Subsequently, the data flow is forwarded with the QoS guarantees.
- Packets that exceed the specifications of the reservation installed is treated as Best-Effort traffic.

1-5 March 1999

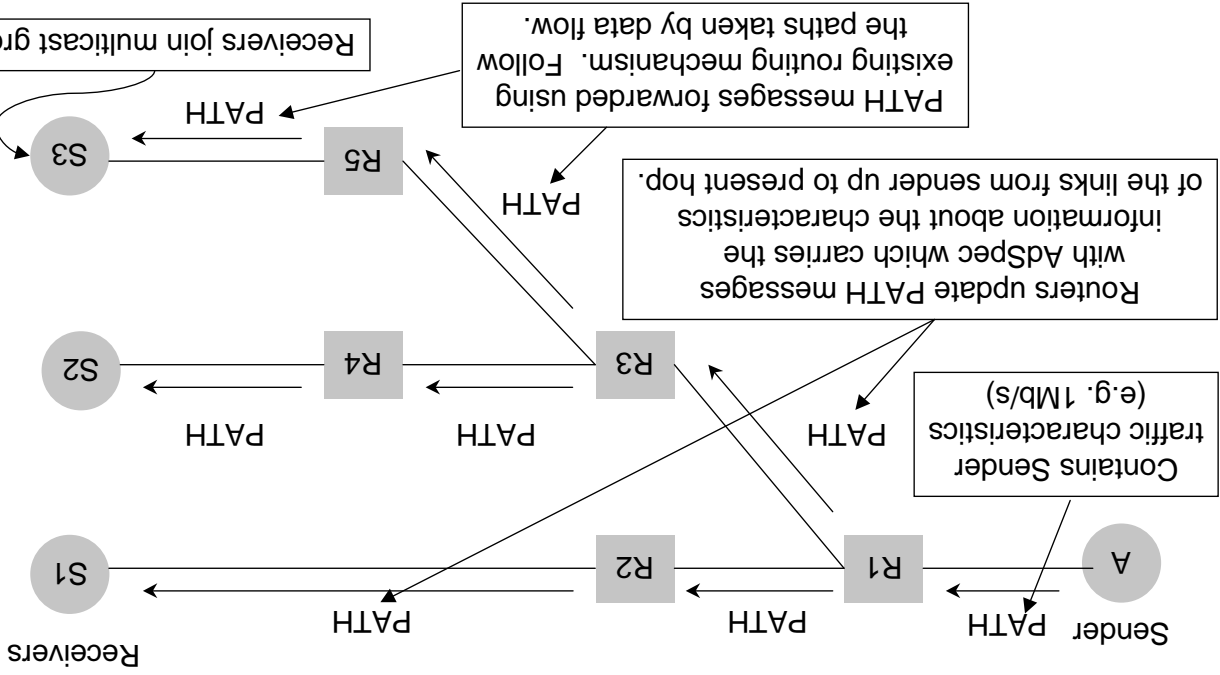
Singapore Linux Conference

18

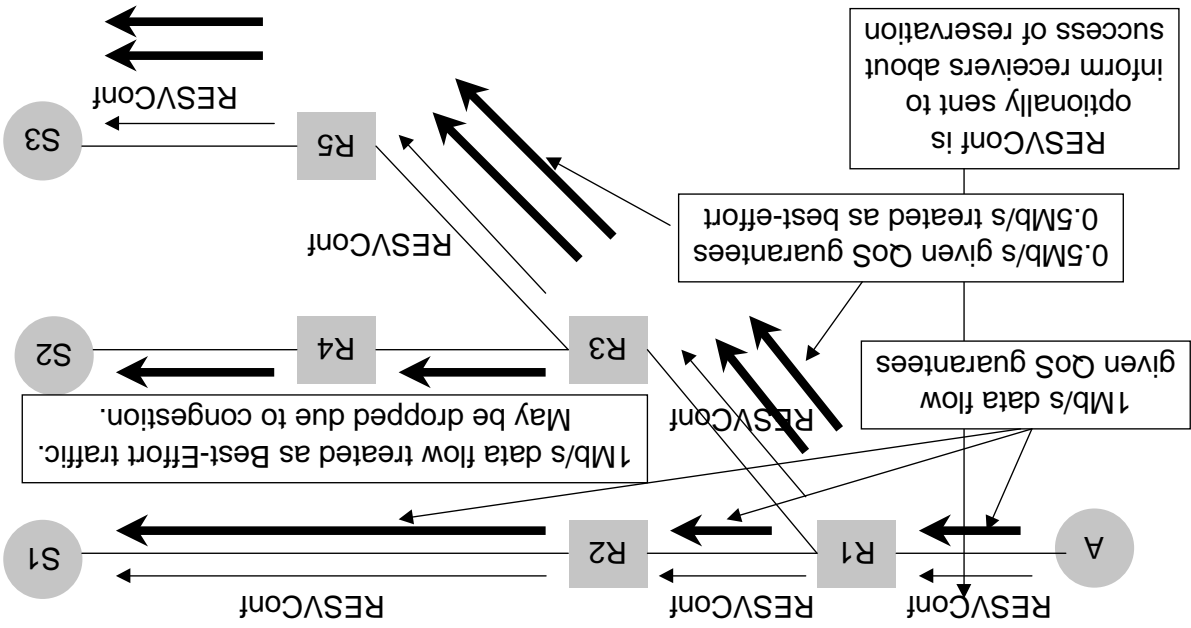
Making Reservation



Making Reservation



Making Reservation



Making Reservation

- PATH and RESV are sent periodically to refresh the states on all nodes along the path.
- If any message is not received within the specified period, the reservation is removed.
- Receivers can alter reservation by sending RESV with new FlowSpec.
- If a particular router cannot meet the reservation request of receivers, RESVErr is sent in the directions of the receivers.

Removing Reservation

- Senders can end a session by sending PATHtear. Similarly, receivers can end a session by sending RESVtear.
- Alternatively, they can stop sending PATH or RESV and let the reservation states times out.

References

- Resource Reservation Protocol (RSVP) WG
 - <http://www.ietf.org/html.charters/rsvp-charter.html>
- RSVP implementation
 - <http://www.isi.edu/div7/rsvp/rsvp.html>
- RSVP Admission Policy (RAP) WG
 - <http://www.ietf.org/html.charters/rap-charter.html>
- Integrated Services (IntServ) WG
 - <http://www.ietf.org/html.charters/intserv-charter.html>
- Integrated Services on Specific Link Layers (ISSLL) WG
 - <http://www.ietf.org/html.charters/issll-charter.html>

References (cont.)

- D. D. Clark, S. Shenker, L. Zhang
 - Supporting Real-Time Applications in an Integrated Services Packet Network: Architecture and Mechanism, *Sigcomm '92*, August 1992.
- L. Zhang, S. Deering, D. Estrin, S. Shenker, D. Zappala
 - RSVP: A New Resource Reservation Protocol, *IEEE Network*, Sept. 1993
- Paul P. White
 - RSVP and Integrated Services in the Internet: A Tutorial, *IEEE Communications Magazine*, May 1997

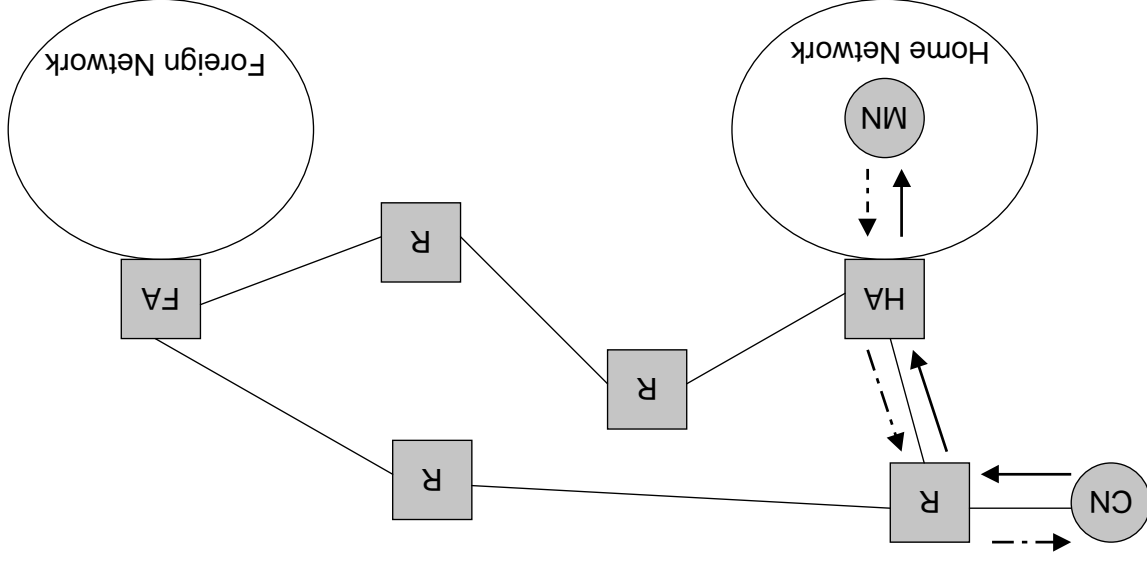
Mobile IP

- A protocol that enables hosts to move from one IP subnet to another and yet:
 - be reachable.
 - maintain existing connections.
- Layer 3 technology that can be used with any link-layer device, whether wired or wireless.

Salient features of Mobile IP

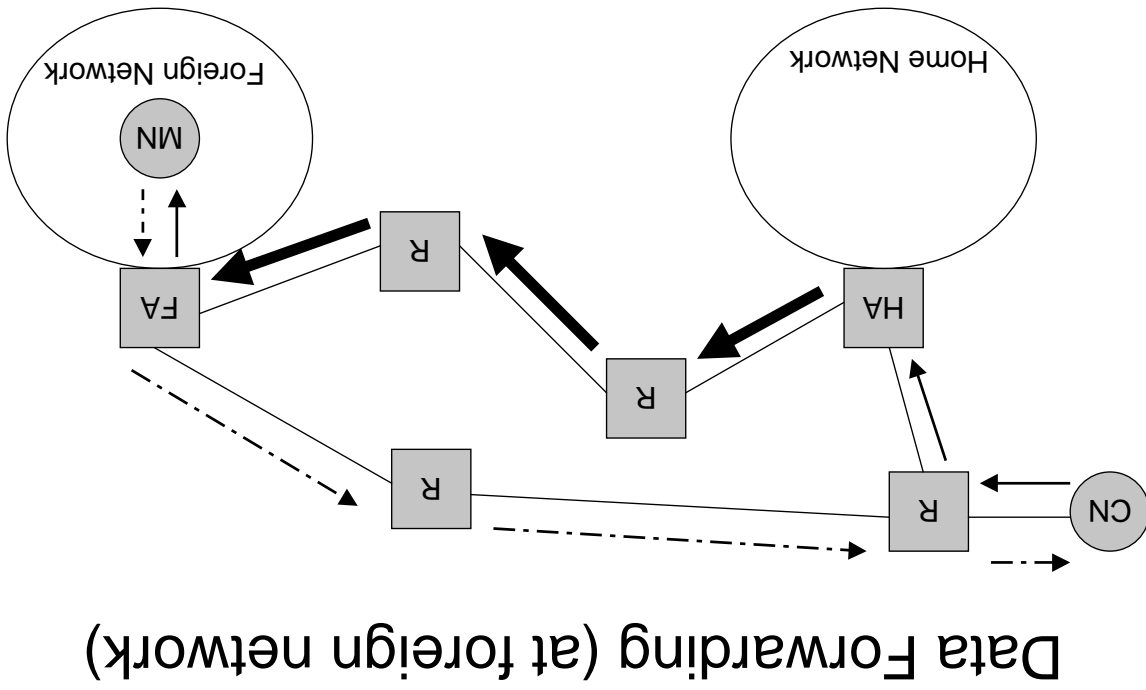
- Mobile nodes (MN) can move from one IP subnet to another.
- Mobility support provided using home agents (HA) and foreign agents (FA).
- Employs protocol tunneling for data forwarding.
- Uses soft-state (time-outs and refreshes).
- Signalling is secure by design (keyed-MD5).
- Changes required only at HA, FA, MN.

Data Forwarding (at home)



- Provides current location and move detection information for mobile nodes.
- HA and FA transmit Agent Advertisements (AA) to advertise their services on a link.
- MN deduces its current location based on the presence of AAs.
- If available, link-layer features can be used for Agent Discovery, thus replacing the use of AAs.

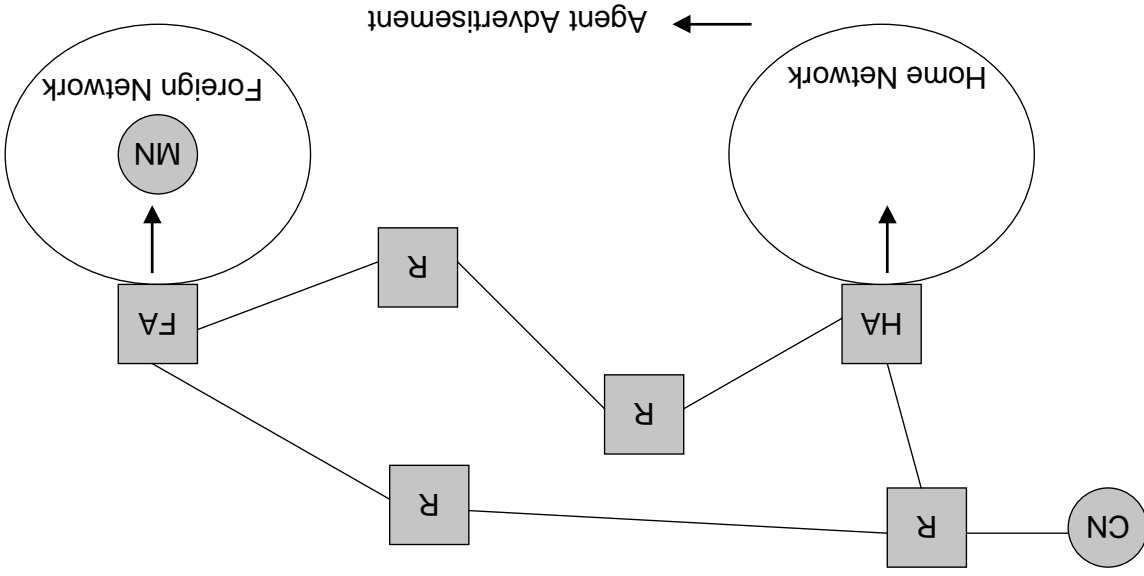
Agent Discovery



Registration

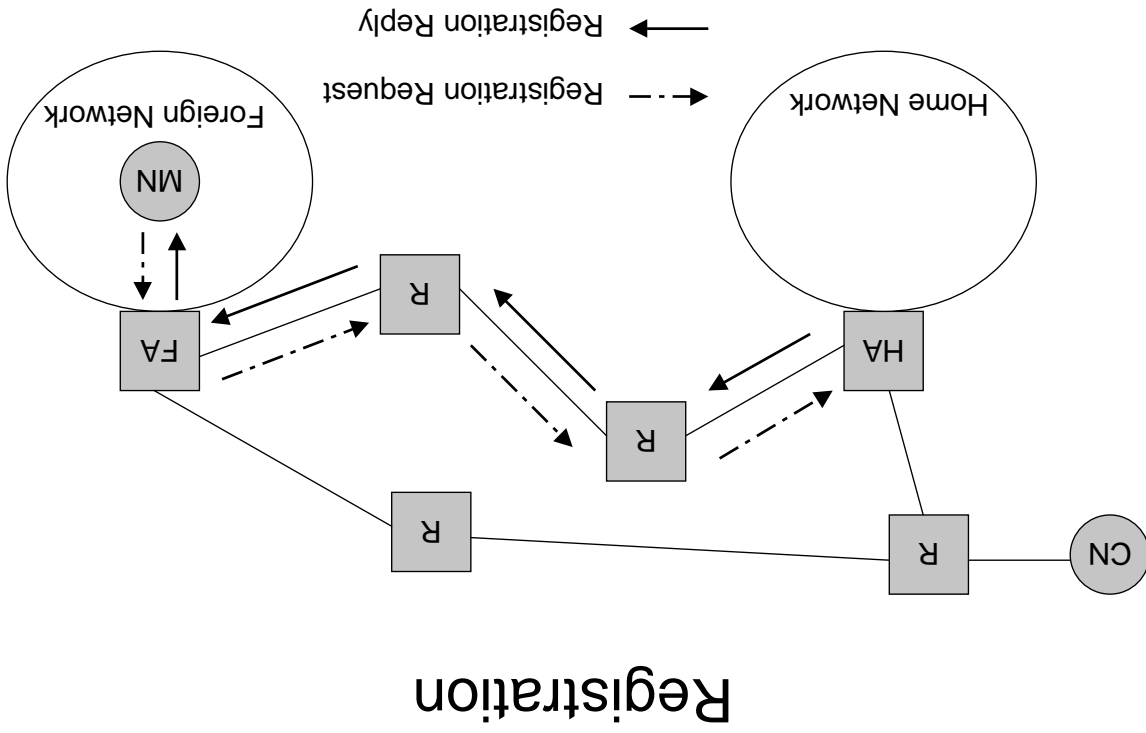
- Used to set up mobility states at HA, FA, MN.
- MN sends Registration Request message to FA.
- FA relays the message to HA.
- HA sets up mobility state and sends up Registration Reply message to FA. FA then sets up mobility state and relays message to MN.
- Mobility states have finite lifetime and require periodic refresh of registration messages to maintain.

Agent Discovery



Security

- Registration message authentication using keyed-MD5.
- MN-HA key is compulsory.
- MN-FA, FA-HA keys are optional.



References

- Mobile IP WG
 - <http://www.ietf.org/html.charters/mobileip-charter.html>
- Mobile IP at NUS
 - <http://mip.ee.nus.edu.sg>
- Charles E. Perkins
 - Mobile IP: Design Principles and Practices, *Addison-Wesley*, 1998
- James D. Solomon
 - Mobile IP: The Internet Unplugged, *Prentice Hall*, 1997

Providing QoS to Mobile Hosts

- The next logical step.
- A vision of networked mobile multimedia in the Internet.

Objectives

- Bring multimedia networking to people on the move.
- Mobile IP telephony?

Issues and Challenges

- Wireless environment eg higher BER, dynamic capacity, blackout regions.
- Effects of mobility on QoS eg handoff latency, call dropping probability.
- Can Mobile IP, RSVP, IntServ be used in this case?

Observations

- RSVP does not inter-operate with Mobile IP:
 - No reservations are made inside the tunnel.
 - RSVP is not aware of mobility of hosts.
- Mobile IP has poor handoff performance:
 - Handoff algorithm not designed for frequent handoffs.
 - Distant registrations between MN and HA increases handoff latency.

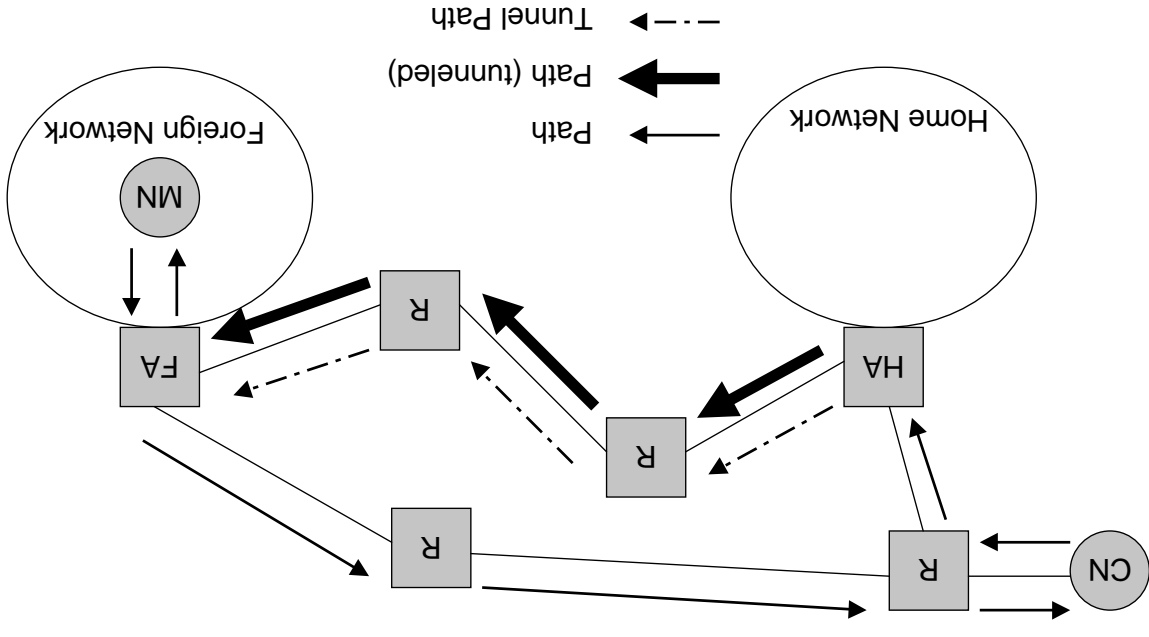
Our Solutions

- Use loose QoS guarantees eg Controlled Load.
- Use Tunnel Support for RSVP extension.
- Inform RSVP of mobility events via Mobile Middleware.
- Improve handoff performance of Mobile IP using fast handoff scheme and an agent hierarchy.
- Improve call dropping probability by using an agent hierarchy and admission control of domain.

Tunnel Support for RSVP

- Recursively apply RSVP over the tunnel portion of the path.
- FF style unicast reservation between the two end points of the tunnel.
- Flows with reservations use UDP encapsulation within the tunnel, instead of the usual IP-in-IP encapsulation.
- Mapping of tunnel session with end-to-end session using SESSION_ASSOC object.

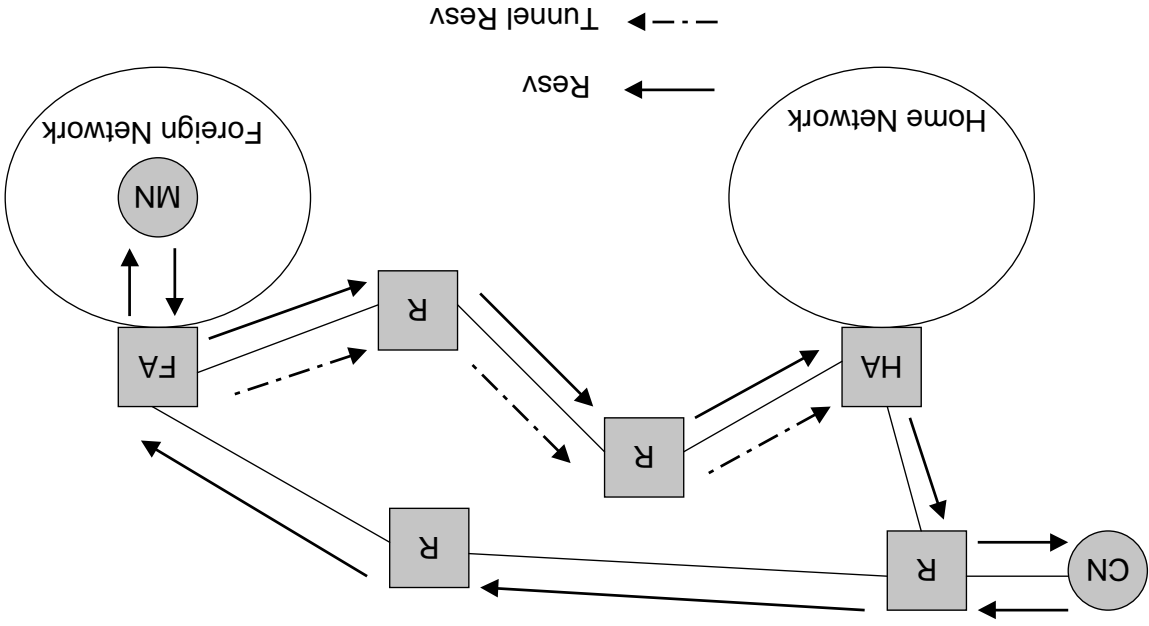
Tunnel Support for RSVP



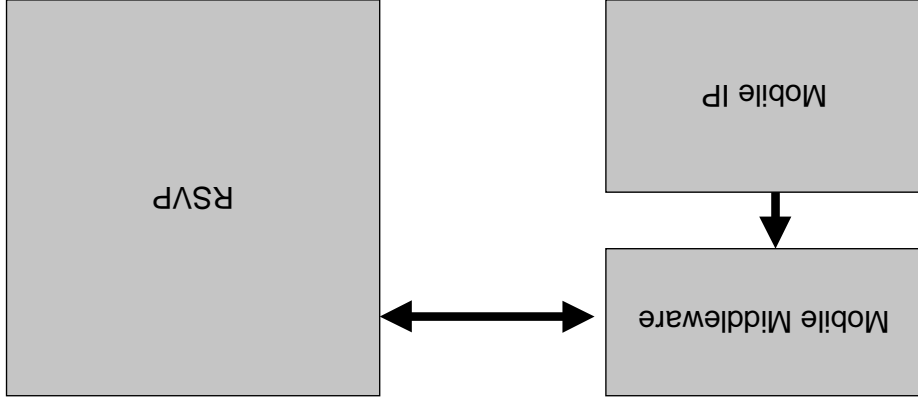
Mobile Middleware

- Layered on top of Mobile IP.
- Provides mobility information to RSV.
- Allows RSV to dynamically set up and tear down RSV tunnel states.
- Reusable by other applications interested in mobility notifications.

Tunnel Support for RSV



Mobile Middleware



Fast Handoff algorithm

- Improves handoff performance of Mobile IP.
- MN registers with an FA as soon as it hears its Agent Advertisements.
- MN has multiple registrations with different FAs in a cell overlap region.
- Seamless handoffs by exploiting the presence of cell overlaps.

Regional Aware Foreign Agent (RAFA)

- Main aim is to reduce registration latency.
- But scheme has other side benefits.
- Concept of an administrative domain.
- Mobility of hosts within an administrative domain is hidden from the home network.
- Mobility within domain handled by Regional Foreign Agents (RFA) and Local Foreign Agents (LFA).

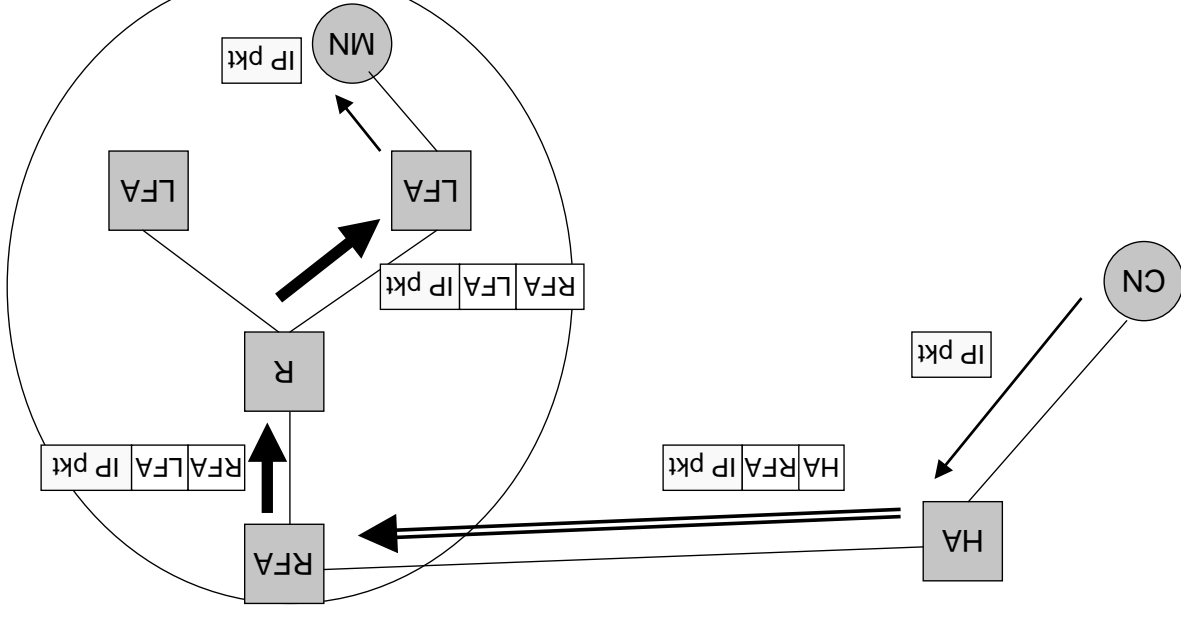
Regional Aware Foreign Agent (RAFA)

- Reduces mobility signalling traffic in the Internet.
- Changes required only at foreign networks. MN and HA can be used unchanged.
- Simplifies billing, traffic and access control into domain.
- Flexible placement of RFAs and LFAs.
- Intermediate routers can be in between.
- Is extendable to more than 2-layers of hierarchy.

Regional Aware Foreign Agent (RAFA)

- Load balancing possible by using multiple RFAs in a domain.
- Autonomous mode capability for intra-domain sessions.
- Key management problem reduced by reducing the number of trusted entities.
- Drawback is RFA requires knowledge of MN-HA key, since it needs to modify the registration messages.

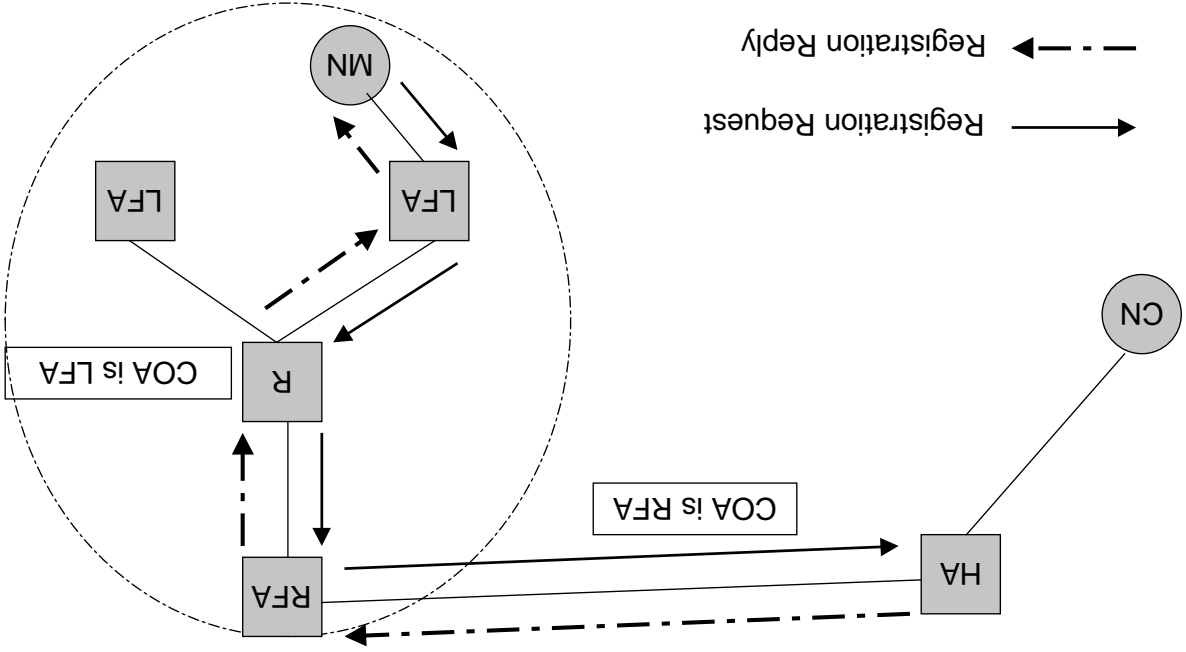
Data Forwarding under RAFA



Admission Control of Domain

- Admission of reserved flows into a domain is controlled.
- Limit the reserved flows going into a domain in order to improve call dropping probability. (at the expense of call blocking probability?)
- Maximize network utilization by allowing best-effort flows to use the rest of the available bandwidth.

Registration under RAFA

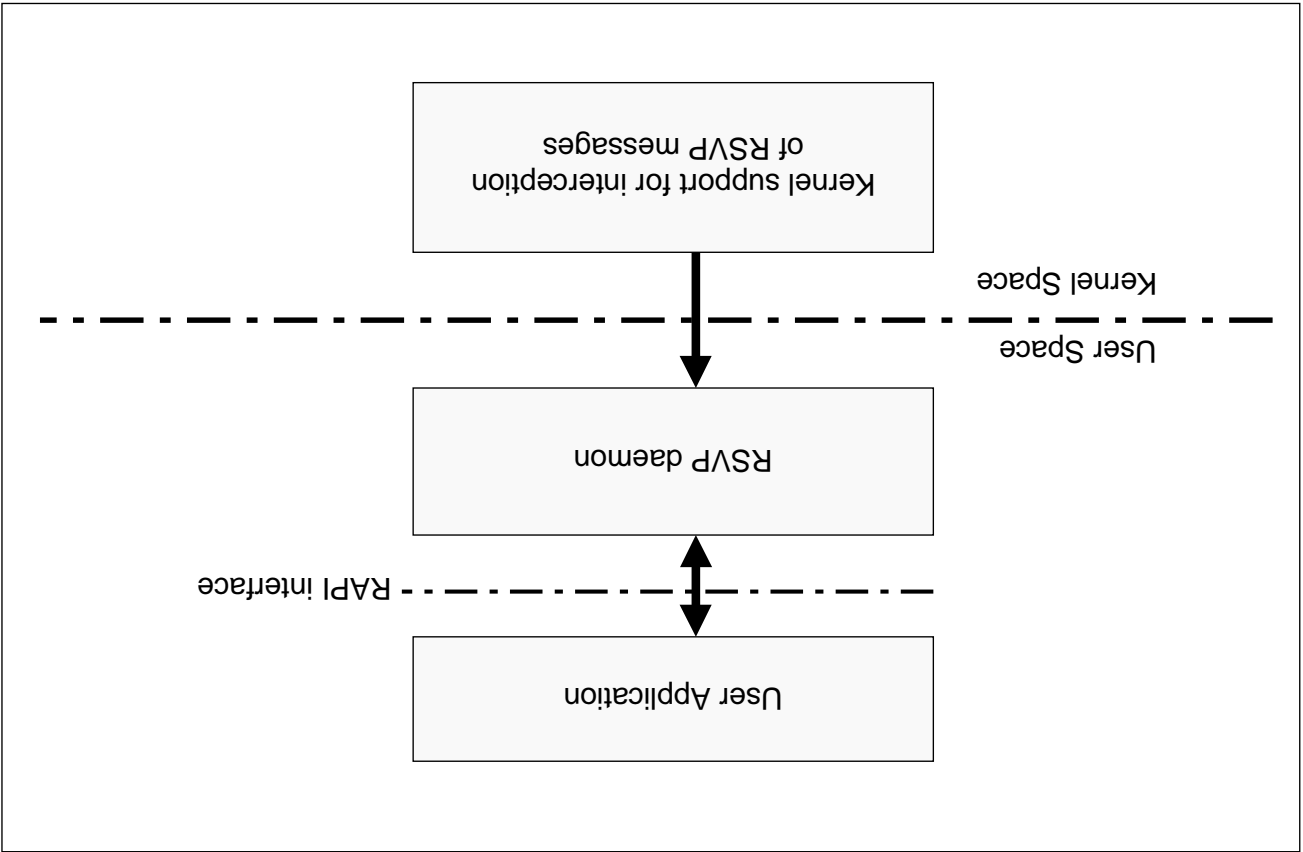
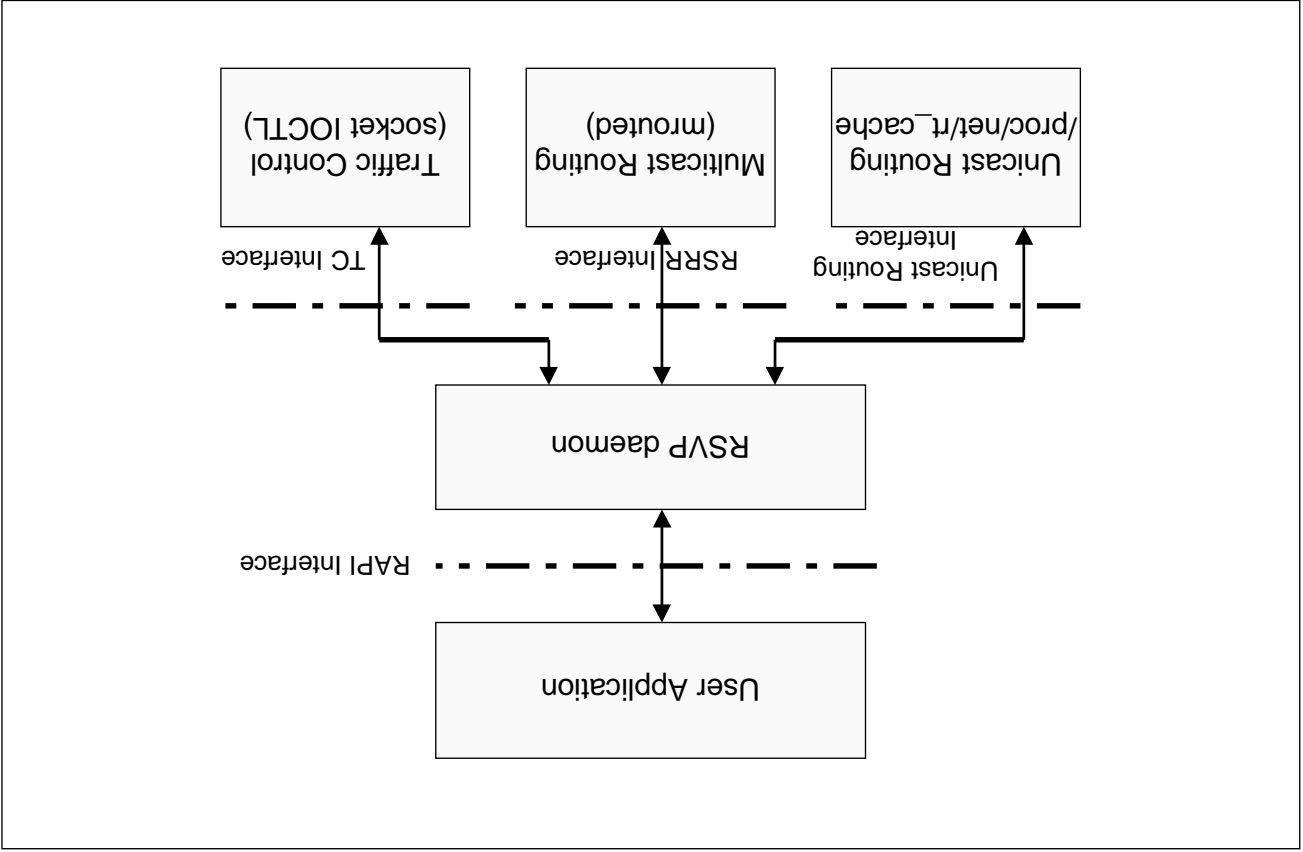


References

- A. Terzis, J. Krawczyk, J. Wroclawski, L. Zhang
 - RSVP Operation Over IP Tunnels, Internet Draft, Work in Progress, August 1998.
- S.F. Foo, K.C. Chua
 - Regional Aware Foreign Agent (RAFA) for Fast Local Handoffs, Internet Draft, Work in Progress, November 1998.
- C.C. Foo
 - TCP Performance in Mobile IP, B.Eng dissertation, National University of Singapore, <http://mip.ee.nus.edu.sg> , 1997.
 - Mobile IP and RSVP, M. Eng thesis, in preparation, 1999.

Implementation of RSVP on Linux

- A port of ISI's RSVP version 4.1.a3 for FreeBSD to Linux.
- RSVP support provided by a user-space program; the RSVP daemon.



RSVP Implementation Details

- Starting point of program
 - rsvp_main.c
- Handling Path, PathTear
 - rsvp_path.c
- Handling Resv, ResvTear
 - rsvp_resv.c
- Handling PathErr, ResvErr, ResvCont
 - rsvp_err.c
- Interface to routing
 - rsvp_rrrr.c, rsvp_unicast.c

RSVP Implementation Details

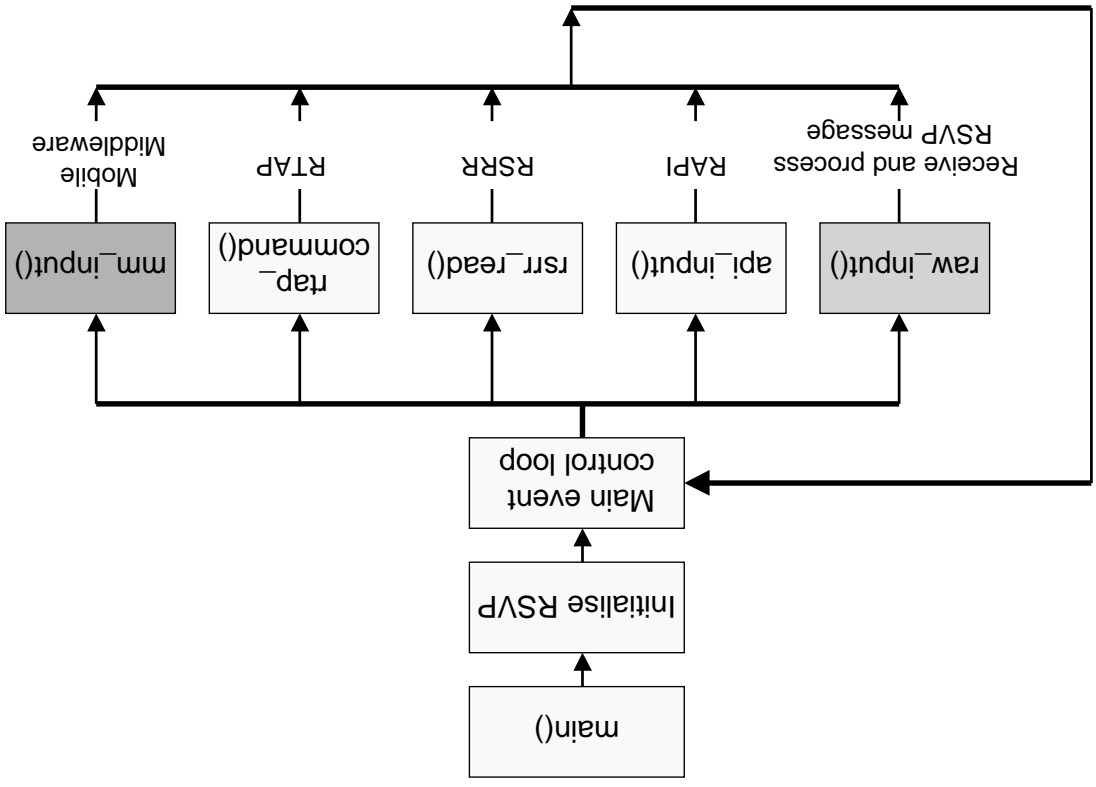
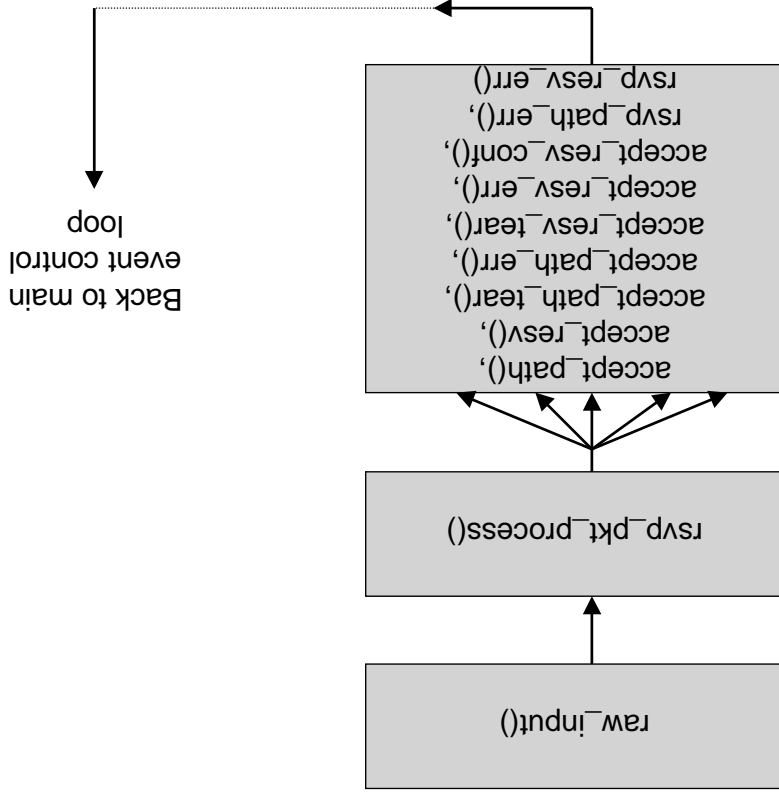
- Interface to traffic control
 - tc_cbq.c, tc_cbqinit.c, tc_test.c
- Status display and debugging
 - rsvp_debug.c, rsvp_mstat.c, rsvp_print.c
- Miscellaneous support
 - rsvp_fm.c, rsvp_lib.c, rsvp_api.c, rsvp_rtap.c
 - rsvp_bord.c, rsvp_specs.c, rsvp_timer.c, rsvp_util.c, rsvp_netio.c, rsvp_md5c.c, rsvp_key.c

Important Data Structures

- Details found in file rsvp_var.h
- Session : Per-session internal data structure.
- PSB : Path State Block.
- RSB : Reservation State Block.
- TCSB : Traffic Control reservation state block.

Structure of RSVP message

- 7 message types (Path, PathTear, PathErr, Resv, ResvTear, ResvErr, ResvConf).
- Raw IP of protid 46, or UDP pkt with ports 1698, 1699.
- Consists of a common RSVP header + a variable no. of objects.
- Objects eg SESSION, RSVP_HOP, FLOWSPEC, FILTER_SPEC,
- Details can be found in file rsvp.h



Implementation Quirks

- IP Router Alert option not implemented.
 - Routers explicitly examine all packets to determine if they are RSVP messages. An RSVP message is identified by an IP protocol id of 46.
- Multicast support for RSVP not implemented.

RSVP Configuration

- Compile kernel for RSVP support:
 - CONFIG_IP_MULTICAST, CONFIG_RSVP
- Compile RSVP daemon.
- Compile rtap (optional).
- Edit configuration file:
 - /etc/rsvpd.conf (optional).

References

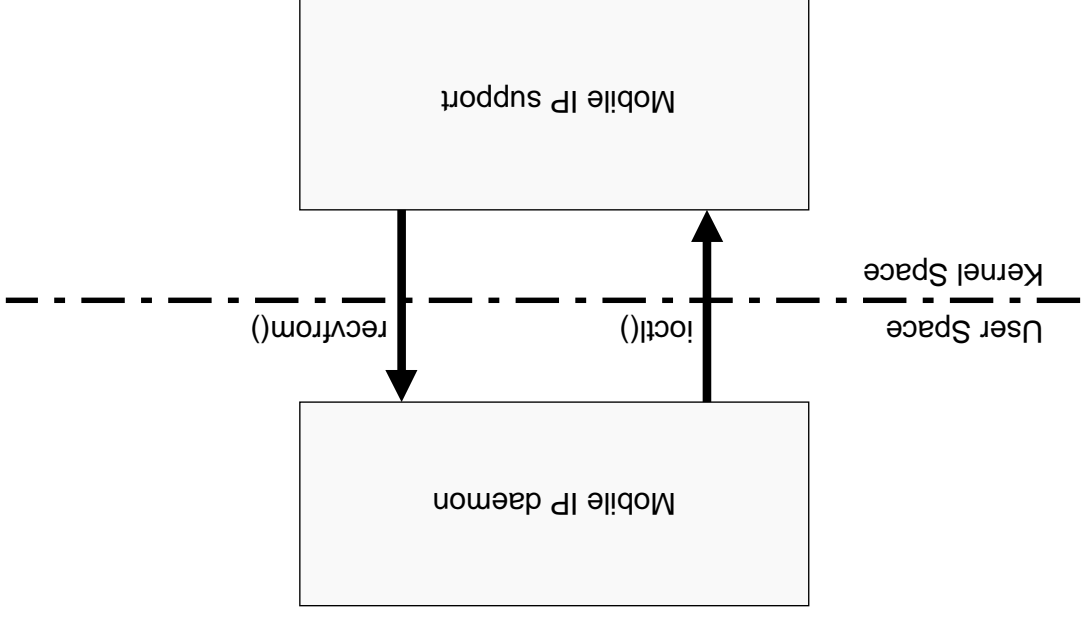
- I. Gui
 - Reservation Protocol and Class Based Queuing on Linux, M.Sc. Dissertation, National University of Singapore, 1998.
- RSVP implementation at ISI
 - <http://www.isi.edu/div7/rsvp/rsvp.html>
- R. Braden, L. Zhang
 - RFC 2209, Resource ReSeRVation Protocol (RSVP) -- Version 1 Message Processing Rules, Sep 1997.
- R. Braden (ed)
 - RFC 2205, Resource ReSeRVation Protocol (RSVP) -- Version 1 Functional Specification, Sep 1997.

Implementation of MIP in Linux

- Combination of user-space and kernel space.
- Bulk of code inside the kernel.
- So far three major versions:
 - version 1.2 - kernel 1.3.55
 - version 2.0beta - kernel 2.0.24
 - version 3.0beta - kernel 2.0.34, bugfixes, additional features (yet to be released to the public)

Features of MIP implementation

- Implemented: (for IPv4)
 - Mobile IP base protocol.
 - Route Optimization.
 - Multiple simultaneous mobility bindings.
 - Bi-tunneling.
 - Multicast support for MIP (bi-tunnel scheme).
 - Regional Aware Foreign Agent (RAFA).
- Not yet implemented:
 - Co-located care-of addresses.



Newly Added Files

- /usr/src/linux/net/ipv4/ :
 - mip.c
 - Shared MIP routines.
 - mip_agent.c - Agents code (HA, FA, RFA, LFA).
 - mip_node.c - Mobile node code.
 - mip_md5.c - RSA MD5 utilities.
 - mip_optim.c - Route Optimization code.
- /usr/src/linux/include/linux/mip.h
- /usr/src/linux/include/net/mip.h
- /usr/src/linux/include/net/mip_optim.h

Modified files

- /usr/src/linux/net/ipv4/ :
 - icmp.c
 - Intercept agent advertisements
 - udp.c
 - Intercept registration messages
 - arp.c
 - MIP processing of ARP
 - ip_input.c - HA forwards traffic to MN
 - ip_output.c - HA forwards traffic to MN
 - route.c, ipip.c, at_inet.c, dev_inet.c
- /usr/src/linux/include/linux/ :
 - icmp.h, proc_fs.h, sockios.h

Important Data Structures

- mip_reg_list - registrations list
- mip_agent_list - agents list
- mip_coaddr_list - COAs list
- mip_key_list - MN-HA key list
- mip_device - extension to dev structure
- mip_router_list - router list
- mip_dcv_global - MIP config options
- Details found in /usr/src/linux/include/net/mip.h

MIP States

- Registration states
 - MIP_REG_STATE2 - deregistered state
 - MIP_REG_STATE3 - registration request state
 - MIP_REG_STATE4 - registered state
 - MIP_REG_STATE5 - registration reset awaiting state
 - MIP_REG_STATE6 - optim binding cache entry state
- Agent Discovery states
 - MIP_DCV_STATE0 - agent unlocated state
 - MIP_DCV_STATE1 - solicitation request state
 - MIP_DCV_STATE2 - agent located state

Major Functions

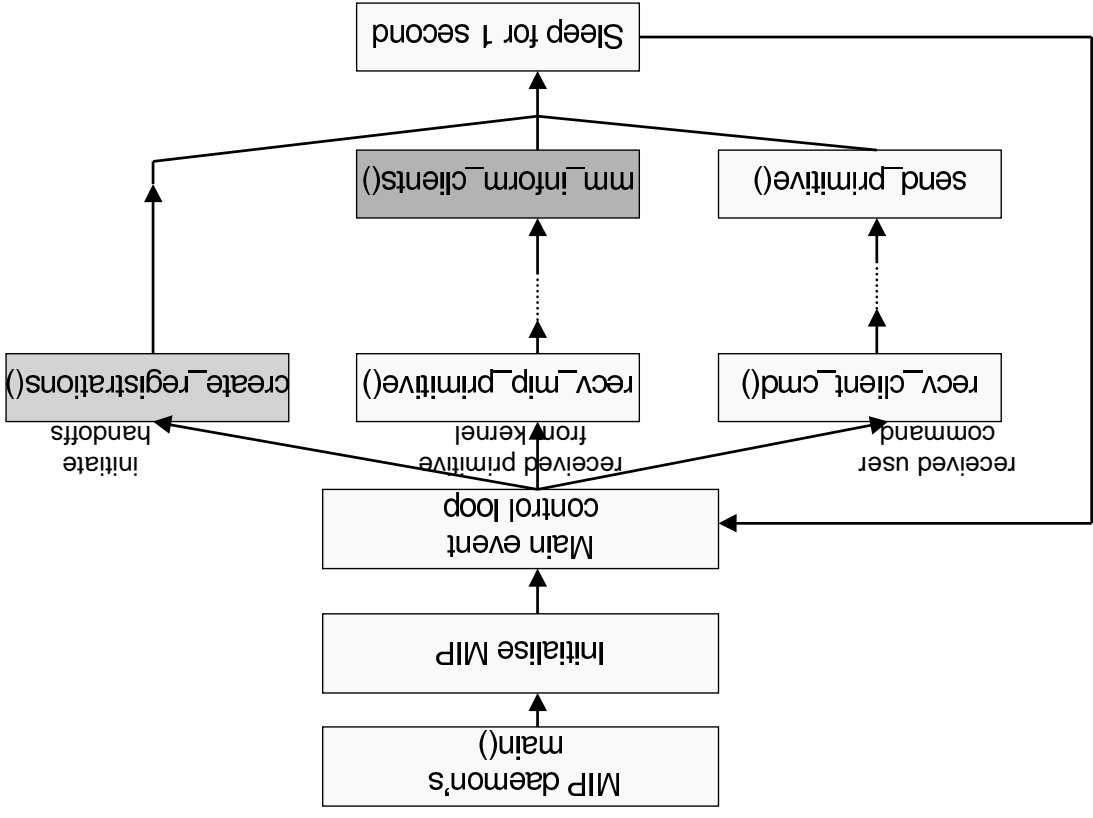
- `mobile_icmp_rcv()`
 - Interception at `icmp_rcv()` in file `icmp.c`
 - Agent Discovery procedure using ICMP message.
- `mobile_signalling_rcv()`
 - Interception at `udp_rcv()` in file `udp.c`
 - Registration procedure using UDP message.
- `mip_seek_tunnels()`
 - An interception in files `ip_input.c` and `ip_output.c`
 - Handles encapsulation of packets to mobile nodes.

Major Functions

- `mip_reg_request_rcv_ha()`,
`mip_reg_request_rcv_fa()`
 - processes reg request messages
- `mip_request_rcv_from_mn()`
 - processes deregistration message from MN
- `mip_reg_reply_rcv()`
 - processes reg reply messages
- `mip_ioctl()`
 - Mobile IP `ioctl()`, intercepted in file `at_inet.c`

Interface to User-space

- Socket IOCTL calls:
 - SIOCONMIP - turn on MIP stack
 - SIOCOFFMIP - turn off MIP stack
 - SIOCREQMIP - send primitive to the mip stack
- Primitives sent to MIP daemon socket:
 - Implemented in kernel in queue_*(*) functions
- Proc filesystem support:
 - /proc/net/mip_dev
 - /proc/net/mip_reg
 - /proc/net/mip_agent (MN only)



Mobile IP Configuration

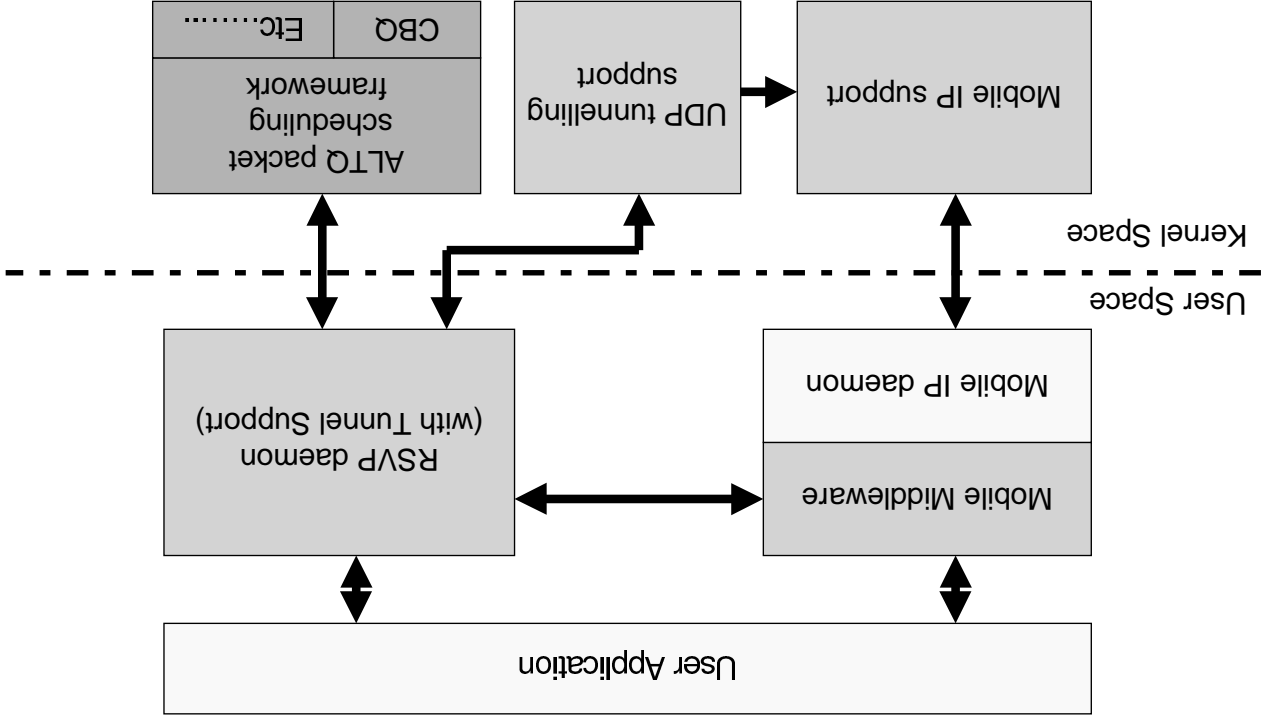
- Compile kernel for Mobile IP support:
 - CONFIG_NET_IPIP, CONFIG_IP_FORWARD
 - CONFIG_MOBILE_IP
 - CONFIG_HOME_AGENT (for HA)
 - CONFIG_FOREIGN_AGENT (for FA)
 - CONFIG_MOBILE_NODE (for MN)
- Compile MIP daemon code.
- Edit configuration files:
 - /etc/agtserv.conf for HA and FA.
 - /etc/mnserv.conf for MN.

References

- Mobile IP at NUS
 - <http://mip.ee.nus.edu.sg>
- C. Perkins (ed)
 - RFC 2002, IP Mobility Support, Oct 1996.

Providing QoS in Mobile Environment

- Implemented:
 - Tunnel Support for RSVP.
 - Mobile Middleware.
 - Fast Handoff scheme for Mobile IP.
 - Regional Aware Foreign Agent.
- Requires Linux 2.0.34 kernel and libcs.



Tunnel Support for RSVP

- Scheme to support reservations inside a tunnel.
- Core component in our mobile QoS scheme.
- RSVP daemon modified to support tunnels.

Newly Added Files

- rsvp_mm.c
 - Processes mobility information provided by Mobile Middleware.
- rsvp_tunnel.c
 - Functions used in Tunnel Support for RSVP.
- rsvp_mm.h
- rsvp_tunnel.h

Important Modified Files

- `rspv_path.c`
 - `accept_path()` - added processing for Tunnel Path msg
 - `kill_PSB()` - added processing for Tunnel PathTear msg
- `rspv_resv.c`
 - `flow_reservation()` - added processing for Tunnel Resv msg
 - `tear_reserv()` - added processing for Tunnel ResvTear msg
- `rspv.h`
 - definition of TUNNELED_FLOWS object.

TUNNELED_FLOWS object

```
typedef struct {
    struct in_addr saddr;
    struct in_addr daddr;
    u_char        protid;
    u_int16_t      sport;
    u_int16_t      dport;
} Tun_Flow_IPv4;
```

All the fields in Tun_Flow_IPv4 refer to the end-to-end session.

```
typedef struct {
    Object_header tunfl_header;
    union {
        Tun_Flow_IPv4 tunfl_ipv4;
        /* Tun_Flow_IPv6 tunfl_ipv6; */
    } tunfl_u;
} TUNNELED_FLOWS;
```

struct tunnel_binding

```
typedef struct {
    struct in_addr saddr;
    struct in_addr daddr;
    u_char protid;
    u_int16_t sport;
    u_int16_t dport;
    flowinfo_ipv4;
}

typedef struct {
    union {
        flowinfo_ipv4 flow_orig_ipv4;
        flow_orig_u;
    }
    union {
        flowinfo_ipv4 flow_tun_ipv4;
        flow_tun_u;
    }
    tun_binding;
}
```

struct mm_tunnel

```
struct mm_tunnel {
    struct in_addr mnaddr;
    struct in_addr faaddr;
    struct in_addr COA;
    struct in_addr haaddr;
    struct in_addr hawaddr;
    u_char node_type;
    u_int16_t flags;
};

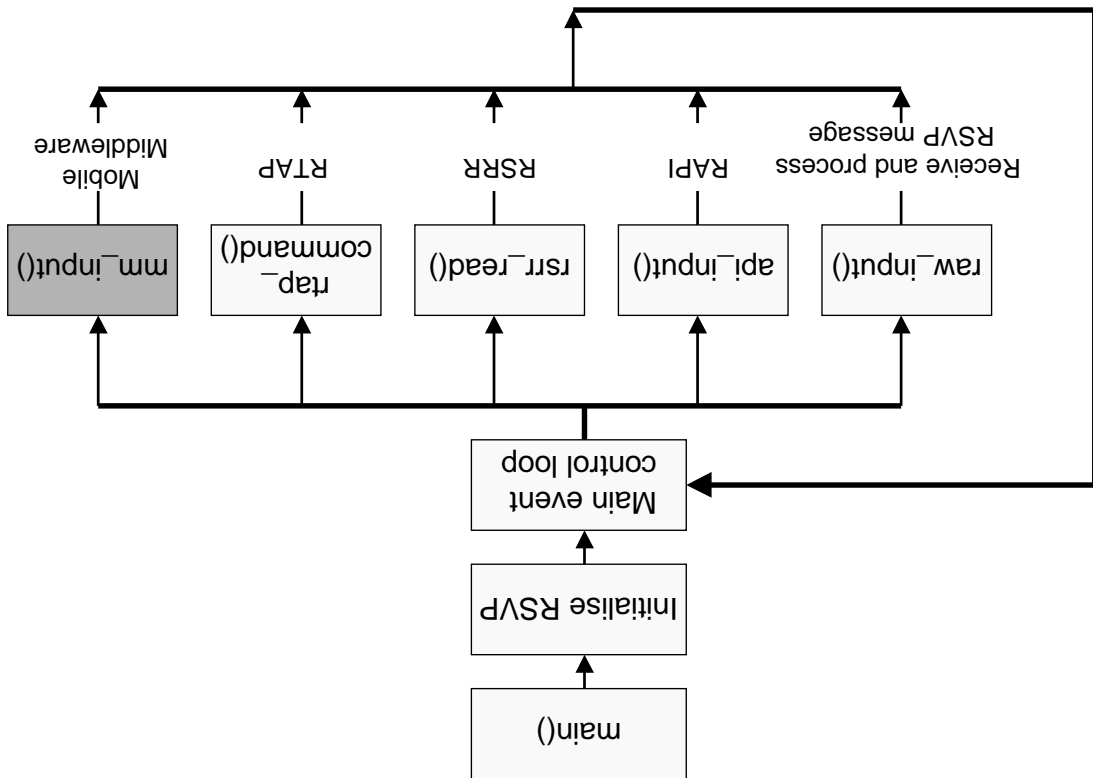
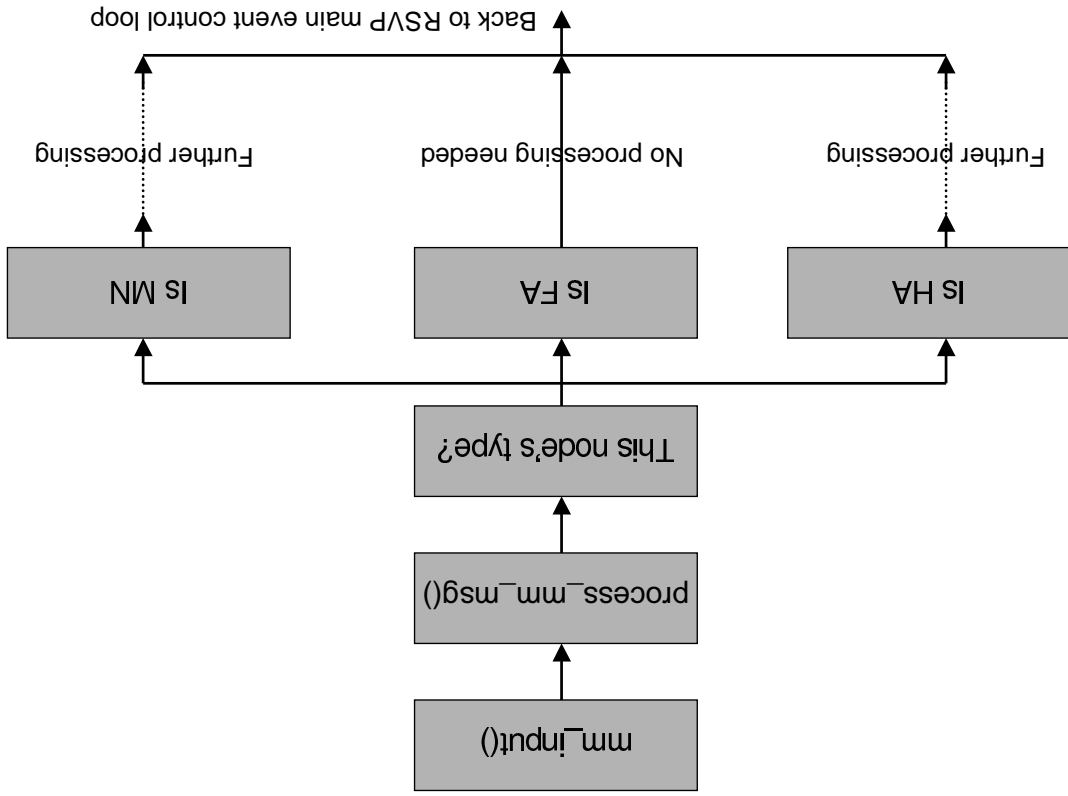
/* Data destined to here */
/* FA wireless address */
/* tunnel end-point */
/* tunnel src-point */
/* HA wireless address */
/* Node type */
/* Other flags */
```

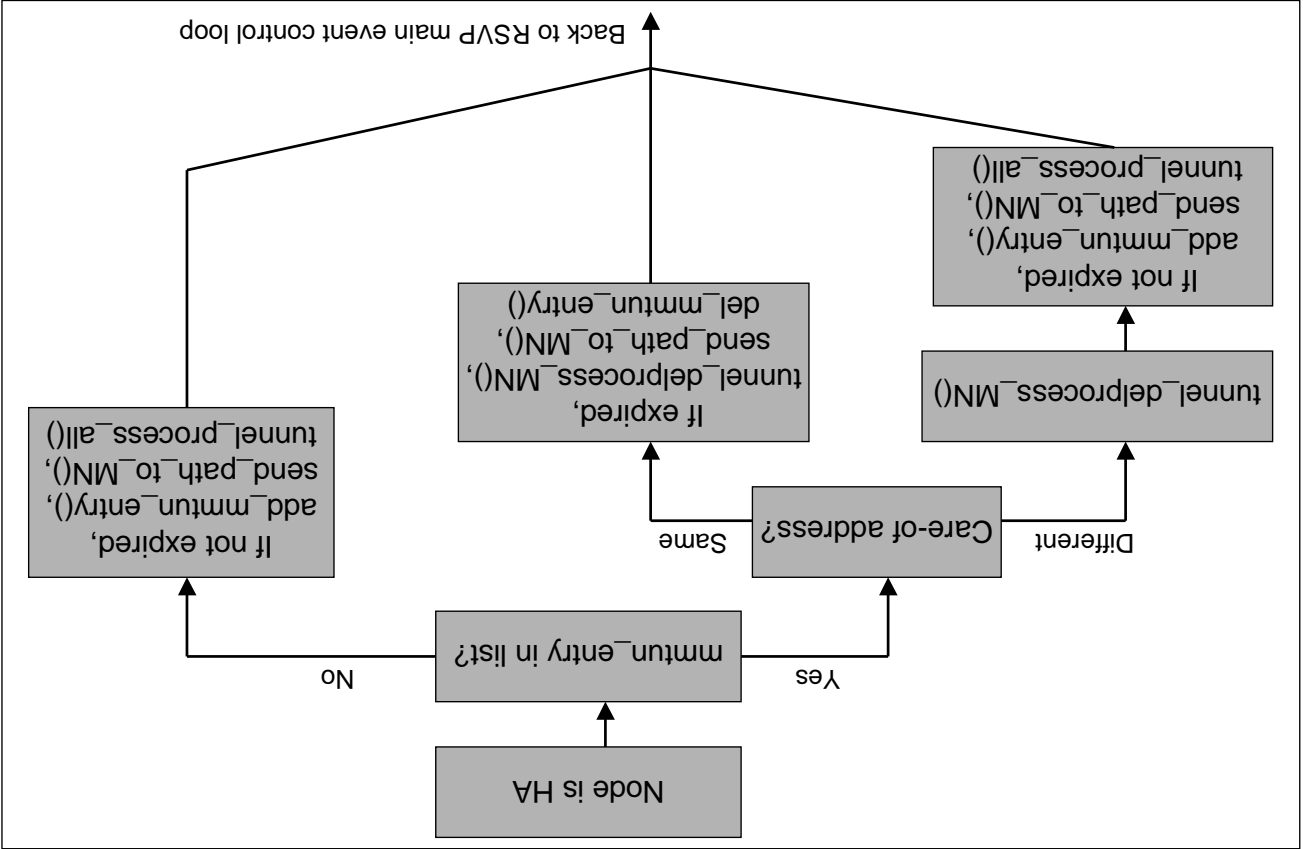
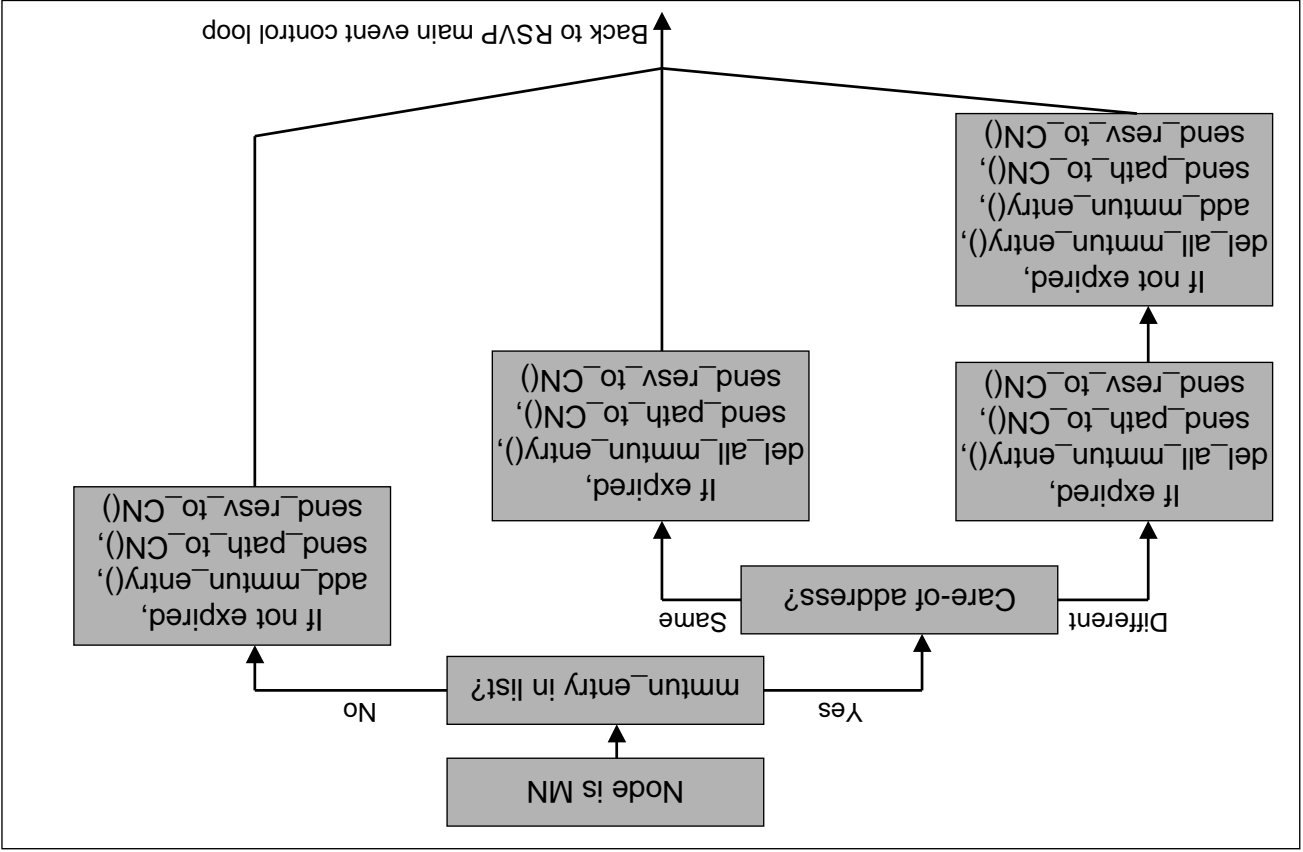
Major Functions

- mm_input()
 - Receives Mobile Middleware messages from MM Server.
- process_mm_msg()
 - Processing of Mobile Middleware messages for RSVP.
- tunnel_process()
 - Processing related to Tunnel Path messages.
- tunnel_process_resv()
 - Processing related to Tunnel Resv messages.

Major Functions

- tunnel_deprocess_MN()
 - Handles removal of tunnel states when mobility binding of MN in HA has expired.
- tunnel_process_pathtear()
 - Processing related to Tunnel PathTear messages.
- tunnel_process_resvtear()
 - Processing related to Tunnel ResvTear messages.





Implementation Quirks

- Implementation differs from latest tunnel specs:
 - UDP encapsulation dest port is 8888, instead of 363.
 - TUNNELED_FLOWS object is used instead of SESSION_ASSOC object.
 - TUNNELED_FLOWS object is placed in the tunnel RSVF messages, while SESSION_ASSOC object is found in the encapsulated end-to-end RSVF messages.

Tunnel Support for RSVF Configuration

- Compile kernel for RSVF support:
 - CONFIG_IP_MULTICAST, CONFIG_UDPENCAP, CONFIG_RSVF
- Compile RSVF daemon (with Tunnel Support):
 - Make sure -DTUNNEL is present in DEFINES section of Makefile.
 - Include -DFASTHANDOFF if you want to support Fast Handoff.

UDP Tunneling Support

- Implemented inside the kernel.
- Used by Mobile IP to determine which flows to be UDP encapsulated.
- RSVF adds/deletes udp_encap entries.

Relevant Files

- /usr/src/linux/net/ipv4/mip.c
- UDP encapsulation of packets in mip_seek_tunnels().
- mip_udp_encap_xmit()
- /usr/src/linux/net/ipv4/udp.c
- Decapsulation of UDP encapsulated packets in udp_rcv().
- /usr/src/linux/net/ipv4/udp_encap.c
- Handling of udp_encap entries.
- /usr/src/linux/include/linux/udp_encap.h
- /usr/src/linux/include/net/udp_encap.h

struct udp_encap

```
struct udp_encap {
    struct udp_encap *prev;
    struct udp_encap *next;
    __u32 saddr;
    __u32 daddr;
    __u16 sport;
    __u16 dport;
    __u8 protocol;
    __u8 active:1;
    __u8 rsvd:7;
    __u16 encapsport;
    /* ip src address */
    /* ip dest address */
    /* ip src port */
    /* ip dest port */
    /* protocol id */
    /* entry is active ? */
};
/* src port to use for udp encap */
```

Interface to User-space

- Socket IOCTL calls :
 - SIOCREQIPINUDP - request UDP encap entry
 - SIOCGACTIPINUDP - activate UDP encap entry
 - SIOCDEACTIPINUDP - deactivate UDP encap entry
 - SIOCDELIPINUDP - delete UDP encap entry
- Proc filesystem support :
 - /proc/net/udp_encap

UDP Tunneling Configuration

- Compile kernel for UDP Tunneling support:
 - CONFIG_UDPECAP
- Compile UDP encapsulation test program (optional).

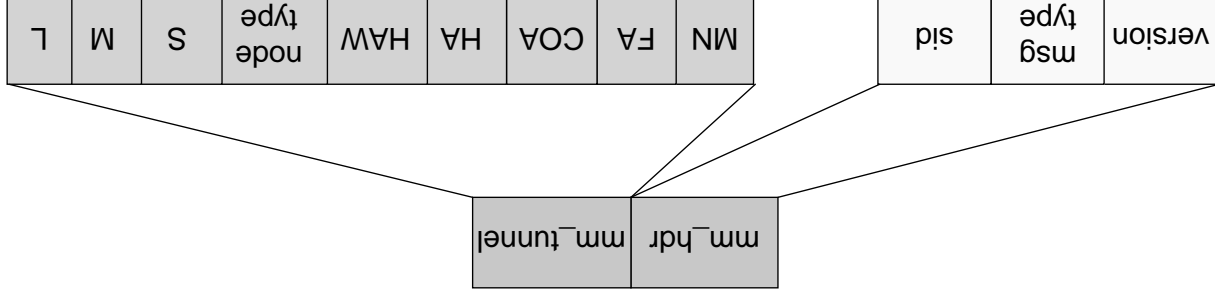
Mobile Middleware (MM)

- MM implemented inside the Mobile IP daemon code, and also in mobile-aware applications.
- Client-server approach. MM Clients request for services of MM Server. MM Server then continuously provides mobility information to MM Clients whenever available.
- Multiple MM Clients can connect to an MM Server.

Mobile Middleware (MM)

- MM Clients and Server communicate with each other via named pipes.
- Uses a standard message format.

MM Message Format



Mobile Middleware Configuration

- Compile MIP daemon code:
 - Make sure CFLAGS has -DMIDDLEWARE included.

Fast Handoff

- Improves handoff performance of Mobile IP.
- MN registers with an FA as soon as it hears its Agent Advertisements.
- MN has multiple registrations with different FAs in a cell overlap region.
- Seamless handoffs by exploiting the presence of cell overlaps.

Modified files

- Code changes needed at the mobile node only.
- MIP daemon's mnserv.c
 - Modified to register with all foreign agents available.
 - /usr/src/linux/net/ipv4/mip_node.c
 - Selective deletion of default routes.

Fast Handoff Configuration

- Compile kernel support for Fast Handoff:
 - CONFIG_MIP_FAST_HANDOFF
- Compile MIP daemon code:
 - Make sure CFLAGS has -DFASTHANDOFF included.
- Edit config file at HA, RFA to support multiple simultaneous mobility bindings.

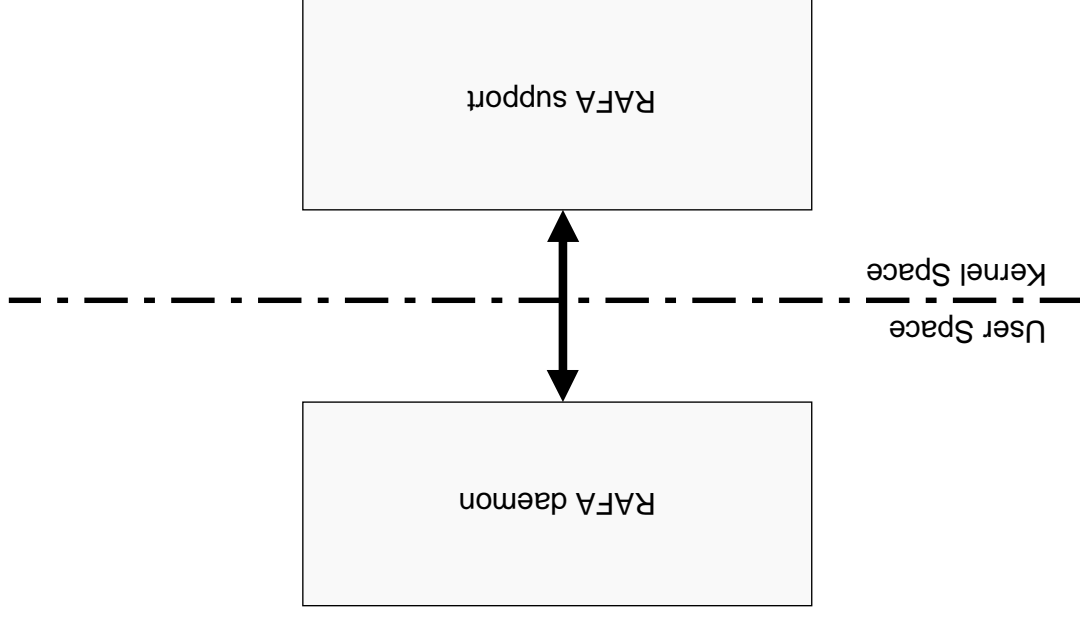
Regional Aware Foreign Agents (RAFA)

- Changes in MIP kernel and daemon code to implement RFA and LFA functionalities.
- Leverages on existing FA code in kernel.
- Minimum changes.

1-5 March 1999

Singapore Linux Conference

107



1-5 March 1999

Singapore Linux Conference

108

Relevant Files

- RAFA daemon
 - ratacli.c, rataserv.c
 - ifaccli.c, ifaserv.c
- /usr/src/linux/net/ipv4/mip.c
 - mip_seek_tunnels()
- /usr/src/linux/net/ipv4/mip_agent.c
 - register_msg_rata_fwd(), register_msg_ifa_fwd(),
 - reg_reply_rata_send(),
 - mip_reg_request_rcv_fa(), mip_reg_reply_rcv()

New Data Structures

- mip_monkey_list
 - MN-HA key list at RFA
- mip_rata_list
 - RFA-LFA list used by RFA
- mip_ifa_list
 - RFA-LFA list used by RFA

RAFA Configuration

- Compile kernel support for RAFA:
 - CONFIG_NET_IPIP, CONFIG_IP_FORWARD
 - CONFIG_MOBILE_IP
 - CONFIG_RAFA
 - CONFIG_LOCAL_FOREIGN_AGENT (for LFA)
 - CONFIG_REGIONAL_AWARE_FOREIGN_AGENT (for RAFA)
- Compile RAFA daemon code.
 - Include the RAFA=y option in the Makefile.

RAFA Configuration

- Edit configuration files:
 - /etc/rafaserv.conf for RAFA.
 - /etc/lfaserv.conf for LFA.

References

- A. Terzis, J. Krawczyk, J. Wroclawski, L. Zhang
 - RSVP Operation Over IP Tunnels, Internet Draft, Work in Progress, August 1998.
- S.F. Foo, K.C. Chua
 - Regional Aware Foreign Agent (RAFA) for Fast Local Handoffs, Internet Draft, Work in Progress, November 1998.
- C.C. Foo
 - Mobile IP and RSVP, M. Eng thesis, in preparation, 1999.

Future Implementation Plans

- Port our MIP code to Linux 2.2 kernels and glibc.
- Upgrade RSVP port to latest ISI version. Also support for glibc.
- New packet scheduling schemes for ALTQ framework. (EDD implementation underway).